PNF Pre-Onboarding / Onboarding & Modeling to Support 5G RAN

• ONAP Modeling
• PNF Onboarding

Benjamin Cheung, PhD
Michela Bevilacqua
Zu Qiang

Jan 08, 2019 version 5 / 3:45PM Pythagore, Nozay, France
Onboarding and Design Time

<table>
<thead>
<tr>
<th>Onboarding Package</th>
<th>NF Descriptor</th>
<th>Platform Model</th>
<th>NF Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre/Onboarding</td>
<td>Pre/Onboarding</td>
<td>Design Time</td>
<td>Run Time</td>
</tr>
<tr>
<td>Vendor</td>
<td>Asset Manager</td>
<td>Service Designer</td>
<td>Operations</td>
</tr>
</tbody>
</table>

**NF Descriptor**
- NF Registration
- PM Dictionary
- PM Schema

**Communication Files**
- Informational Artifacts
- Configuration Files

**PNF Onboarding Package**

**Current Platform Data Model Vs. Target Platform Data Model**

**SDC Catalog**

**A&AI**

**PNF Instance**

**TOSCA**
- Root Node
- Service
- Network Function
- Virtual Link
- Connect Point
- PNF Device
- Complex
- NF Comp

**Ports / NICs**

**Connectivity**
1. **PNF Package Delivery**: Vendor creates & delivers PNF Package with PNF artifacts

2. **PNF Pre-Onboarding** (optional): VNF-SDK (⚠️) can create or validate PNF Onboarding Package

3. **PNF Onboarding**: PNF Package is loaded

4. **Design Time Activities**: NF Onboard Package > SDC catalog

5. **ONAP RT Components**: Ingest and use CSAR package

---

**PNF Pre-Onboarding U/C Overview**

- **PNF Package Delivery**: Vendor creates & delivers PNF Package with PNF artifacts.
- **PNF Pre-Onboarding** (optional): VNF-SDK (⚠️) can create or validate PNF Onboarding Package.
- **PNF Onboarding**: PNF Package is loaded.
- **Design Time Activities**: NF Onboard Package > SDC catalog.
- **ONAP RT Components**: Ingest and use CSAR package.
## Dublin Development Status

<table>
<thead>
<tr>
<th>Component</th>
<th>Impacts Description</th>
<th>JIRA Tickets</th>
</tr>
</thead>
</table>
| SDC       | **PNF Package Format**  
Identify current VNF package format constraints in SDC  
Propose a PNF Package format  
Evaluate VNF and PNF package alignment  
**Package Mapping** - Onboarding PNF package to internal PNF package mapping  
**PNFD Mapping** - Onboarding PNFD to internal PNFD mapping  
A new flow to map ETSI SOL001 PNFD in SDC AID model.  
**Artifact Management**  
Design-time catalog update to associate artifacts to a PNF  
| VNF SDK   | **Package Verification** - PNF package format verification  
**PNF Package Format**  
| Modelling | **PNFD Mapping** – Mapping Onboarded PNFD to platform PNFD mapping  
SDC SW task but PNFD mapping & part of ONAP Resource Data Model. | |
| VNF Requiremets | **Package, Descriptor, Artifacts** - VNF requirements to cover PNF package contents and directory structures and mandatory elements.  
**PNF Descriptor** - Requirements for the PNF Descriptor  
**Artifacts** - Requirements for Artifacts of PNF and PNF Package | VNFRQTS-506, VNFRQTS-496, VNFRQTS-497, VNFRQTS-498, VNFRQTS-499, VNFRQTS-505, VNFRQTS-507, VNFRQTS-508 |
| RT Comp   | Grant backward compatibility use cases on ONAP RT Components | |
PNF PACKAGE CREATION & DELIVERY
PNF ONBOARDING PACKAGE

PNF-D

NF Descriptor

Manifest File

TOSCA Metadata

PF Registration

PM Dictionary

PM Schema

VES Event Registration

Ansible Playbooks

Netconf Yang model

Chef Cookbooks

Manuals, Help files

CustDoc Products

Test files

Licensing agreement

Resource Configuration Info

Informational Artifacts

Communication Files

PNF Onboarding Package

TOSCA Metadata (SDC)

TOSCA Descriptor (SDC)

X_License Model files

SDC Added Files

SDC CSAR Package

SDC Design Time

Vendor Provided

PNF Package Artifacts

Definition Files

PNF Package Artifacts

Configuration Files
PNF Onboarding Package (CSAR)

Note:
- Package Example
- Not all files are listed.
- Folder / file name in blue is requested by SOL004.
- Folder / file name in black is example only.

ROOT

- TOSCA-Metadata
- Definitions
- Artifacts
- MainServiceTemplate.mf

metadata:
- pnf_product_name: gNB
- pnf_provider_id: Ericsson
- pnf_package_version: 1.0
- pnf_release_date_time: 2018-12-03T08:44:00-05:00
- non_mano_artifact_sets:

  Events:
  - source: Artifacts/Deployment/Events/VES_registration.yaml

NF descriptor

- TOSCA.meta
- MainServiceTemplate.yaml

Images
- Deployment
- Scripts
- Tests
- ChangeLog.txt
- MainServiceTemplate.cert
- License_term.txt

Configuration
- Yang_module
- Measurements
- Events
- ...

Install.csh
- ...

pm-dictionary.yaml

yang-module.yang

ves-dictionary.yaml

Not supported by ONAP Casablanca

TOSCA-Meta-Version: 1.0
CSAR-Version: 1.1
Created-By: Ericsson (Zu Qiang 2018-12-03)
Entry-Definitions:
  Definitions/MainServiceTemplate.yaml
Entry-Manifest: MainServiceTemplate.mf
Entry-Change-Log: Artifacts/ChangeLog.txt
Entry-Tests: Artifacts/Tests
Entry-Certificate: Artifacts/License_term.txt

• Package Example
• Not all files are listed.
• Folder / file name in blue is requested by SOL004.
• Folder / file name in black is example only.
metadata: 

# New key Name to be supported by SDC
pnf_product_name: gNB
pnf_provider_id: Ericsson
pnf_package_version: 1.0
pnf_release_date_time: 2018-12-03T08:44:00-05:00

# non mano artifact sets to be supported by SDC
non_mano_artifact_sets:
Events:
  source: Artifacts/Deployment/Events/VES_registration.yaml
Measurements:
  source: Artifacts/Deployment/Measurements/PM_Dictionary.yaml
Yang_module:
  Source: Artifacts/Deployment/Yang_module/Yang_module.yaml
Others:
  Source: Artifacts/Informational/scripts/install.sh
  Source: Artifacts/Informational/user_guide.txt
  Source: Artifacts/Other/installation_guide.txt
  Source: Artifacts/Other/review_log.txt
R4: Modelling VES Events

PNF ONBOARDING FILES

TOSCA-Meta-File-Version: 1.0
CSAR-Version: 1.1
Created-By: Ericsson
Entry-Definitions: Definitions/resource-Ericssongnodeb-template.yml
Entry-Manifest: resource-Ericssongnodeb-template.mf
Entry-Certificate: Artifacts/resource-Ericssongnodeb-template.cert
Entry-Tests: Artifacts/Tests
Entry-Licenses: Artifacts/Licenses/license_term.txt
Entry-Change-Log: Artifacts/ChangeLog.txt
SDC CSAR Package

CSAR file

ROOT
- TOSCA-Metadata
  - Definitions
  - Artifacts
    - MainServiceTemplate.yaml
    - MainServiceTemplate.mf
- TOSCA.meta
- MainServiceTemplate.yaml
- Images
  - MainServiceTemplate.mf
  - Images
- NF_ONBOARDED_PACKAGE
  - Other
  - VFC[name]
    - Tests
      - ChangeLog.txt
      - MainServiceTemplate.cert

Not supported by ONAP Casablanca

- HEAT
- HEAT_VOL
- HEAT_NET
- HEAT_ENV
- HEAT_ARTIFACT
- HEAT_NESTED
- YANG_XML
- MODEL_INVENTORY_PROFILE
- VNF_CATALOG
- VNF_LICENSE
- VENDOR_LICENSE
- APPC_CONFIG
- DCAE_TOSCA
- DCAE_JSON
- PLAN

Directory list under review

Copy of NF descriptor created by SDC

Not supported by ONAP Casablanca

Directory list under review

VFC[name]/Deployment
VFC[name]/Information (can we move it in Deployment dir ?)

Any unrecognized artifacts

Guide
- Install.csh
- ...

THE LINUX FOUNDATION

ONAP
PNF ONBOARDING: SDC Catalog

3 PNF Onboarding: PNF Package is loaded

PNF-D

NF Descriptor

VES Event Registration Specification

PM Dictionary
PM Schema

Manuals, Help files
CustDoc Products
Test files

Licensing agreement

Informational Artifacts

PNF Registration

Configuration Info

PNF Package creation

Ansible Playbooks
Netconf Yang model
Chef Cookbooks

Communication Files

WinZIP
Validating Content

xNF SDK

SDC

PM Dictionary
PM Schema

SDC Catalog

Ansible Playbooks
Netconf Yang model
Chef Cookbooks

SDC Catalog

Ansible Playbooks
Netconf Yang model
Chef Cookbooks

SDC Catalog

Ansible Playbooks
Netconf Yang model
Chef Cookbooks
**xNF ONBOARDING Dependencies R3**

- **Pre-onboarding**
  - VNF SDK
  - SDC
  - Vendor VNF CSAR
  - VNFD ETSI SOL001 based

- **Onboarding Design Time**
  - VNF SOL001
  - Heat template
  - VNF heat template
  - VNF SDK
  - SDC
  - Vendor VNF CSAR
  - VNFD ETSI SOL001 based

- **TOSCA AID VNF**

- **SO**

- **PNF defined by UI**

- **SDNC/APPCC Multi-VIM**

- **VFC**

- **VOLTE workflow**

- **vCPE CreateVcpeWorkflow**

- **vCPE Create&ActivateWorkflow**

- **AID**

- **TOSCA AID PNF**

- **AID**
DESIGN TIME & SDC
NFV Release 2: stage 2 and stage 3 specification summary

(*) Release 2 Stage 3 work items in “green”

- NFV-IFA 015 (NFV Information Model Report)
- NFV-IFA 016 (Papyrus Guidelines)
- NFV-IFA 017 (UML Modeling Guidelines)
- NFV-IFA 024 (NFV Information Model External Touchpoints)

NFV-IFA 003 (Acceleration)
NFV-IFA 004 (Acceleration)
NFV-IFA 006
NFV-IFA 007
NFV-IFA 010 (NFV-MANO Functional Reqs)
NFV-IFA 011 (VNF Pkg)
NFV-IFA 013
NFV-SOL005
NFV-SOL 002
NFV-SOL 003
NFV-SOL 004 (VNF Packaging)
NFV-SOL 006 (VNF and NS Descriptors)
NFV-SOL 007 (NSD file structure)
NFV-SOL 008
NFV-IFA 001
PNF Descriptor impacts
PNF package impacts
Published!
Published!
**DsnTm: PNFD > ONAP Platform Model**

**Design Time Activities:** NF
- Onboard Package > SDC catalog

**PNFD DESCRIPTOR**
- **PNF**
  - Properties:
    - **NODE**
    - DependsOn
    - PnfExtCp
      - NODE
      - VirtualLinkable

**PLATFORM MODEL (INTERNAL MODEL)**
- **TOSCA Root Node**
- **Service**
- **Connection Point**
- **PNF Device**
- **Network Function**
- **Virtual Link**
- **Complex**
- **NF Component**

**LEGEND:**
- PNFD DESCRIPTOR
- Ns Virtual Link
- VirtualLinkable
- NODE
- Connectivity

**Requirements**
- Ports / NICs

---

**THE LINUX FOUNDATION**

---

**ONAP**
Design-Time Process

**DESIGN-TIME (SDC)**
- Policy Designer
- Workflow Studio
- CLAMP Designer
- Service Designer
- DCAE Design Studio
- Add Policies
- Add Workflows
- Add Control Loops
- Define Services
- Define Templates
- Onboarding Package
- Vendor MetaData
- SDC Catalog

**ONAP RUN-TIME**
- Certification Studio
- Deployment Studio
- SDC Distribute CSAR
- DCAE Analytics
- SO
- SDC
- Controller
- A&AI Inventory, Listener
- NF Onboarding Package
- NF Vendor MetaData
- Policy Database
Defining a Service

Design Time Activities:
- Designing a Service
- Distribute
- SDC CSAR Package

SERVICE

- Informational Artifacts
  - Configuration Files
    - Vendor Provided
  - Deployment Artifacts
    - Heat Templates
    - Connection Point
    - Virtual Link
    - Foundational
- Policies
  - Templates
  - Blueprints
    - User Designed
- Resources
  - VNF
  - PNF
  - ANF (Allotted)

THE LINUX FOUNDATION
VNF Requirements

NF Descriptor
Onboarding
Asset Manager
VNF Requirement extension for PNF

- **VNFRQTS-506**: Supporting PNF package onboarding
  - **VNFRQTS-507**: Project scope to include PNF
  - **VNFRQTS-508**: Add PNFD requirements
  - Section 5.1.6:
    - **VNFRQTS-499**: PNF onboarding CSAR package structure based SOL004
    - **VNFRQTS-497**: Adding package security requirements
  - Section 7.2:
    - Clarifications on the documentation requirements
    - **VNFRQTS-505**: PNF onboarding package artifacts
    - **VNFRQTS-498**: Add VES Event Registration requirement to PNF package
    - Clarifications on artifacts structure requirements
    - **VNFRQTS-496**: supporting Ansible protocol in PNF
## ONAP VNF Descriptor (5.1.9) Requirements

<table>
<thead>
<tr>
<th>R number</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-35854</td>
<td>The VNF Descriptor (VNFD) provided by VNF vendor <strong>MUST</strong> comply with TOSCA/YAML based Service template for VNF descriptor specified in ETSI NFV-SOL001.</td>
<td>Shall applicable to PNFD</td>
</tr>
<tr>
<td>R-65486</td>
<td>The VNFD <strong>MUST</strong> comply with ETSI GS NFV-SOL001 document endorsing the above mentioned NFV Profile and maintaining the gaps with the requirements specified in ETSI GS NFV-IFA011 standard.</td>
<td>Shall applicable to PNFD</td>
</tr>
<tr>
<td>R-17852</td>
<td>The VNFD <strong>MAY</strong> include TOSCA/YAML definitions that are not part of NFV Profile. If provided, these definitions <strong>MUST</strong> comply with TOSCA Simple Profile in YAML v.1.2.</td>
<td>Shall applicable to PND</td>
</tr>
<tr>
<td>R-46527</td>
<td>A VNFD is a deployment template which describes a VNF in terms of deployment and operational behavior requirements. … including topology, deployment aspect, and VNF lifecycle management (LCM) operations</td>
<td>PNF LCM is not defined yet</td>
</tr>
<tr>
<td>R-15837</td>
<td>The major TOSCA Types specified in ETSI NFV-SOL001 standard draft</td>
<td></td>
</tr>
</tbody>
</table>
| R-54356  | VNF Data Types | CpProtocolData  
AddressData  
L2AddressData  
L3AddressData  
LocationInformation  
CivicAddressElement |
| R-54876  | | |
| R-67895  | VNF Capability Types | VirtualLinkable |
| R-95321  | VNF Relationship Types | VirtualLinksTo |
| R-32155  | VNF Interface Types | PNF Node Types: PNF, PnfExtCp, Cp |
|          | | PBF Policy Types |

---

**VNF Data Types**

- CpProtocolData
- AddressData
- L2AddressData
- L3AddressData
- LocationInformation
- CivicAddressElement

**VNF Capability Types**

- VirtualLinkable

**VNF Relationship Types**

- VirtualLinksTo

**VNF Interface Types**

- PNF Node Types: PNF, PnfExtCp, Cp
- PBF Policy Types
## ONAP VNF CSAR Package (5.1.6) Requirements

<table>
<thead>
<tr>
<th>R number</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-51347</td>
<td>The VNF package <strong>MUST</strong> be arranged as a CSAR archive as specified in TOSCA Simple Profile in YAML 1.2.</td>
<td>Shall applicable to PNF package</td>
</tr>
<tr>
<td>R-87234</td>
<td>The VNF package provided by a VNF vendor <strong>MAY</strong> be either with TOSCA-Metadata directory (CSAR Option 1) or without TOSCA-Metadata directory (CSAR Option 2) as specified in ETSI GS NFV-SOL004. On-boarding entity (ONAP SDC) must support both options. <strong>Note:</strong> SDC supports only the CSAR Option 1 in Casablanca. The Option 2 will be considered in future ONAP releases.</td>
<td>Shall applicable to PNF package</td>
</tr>
<tr>
<td>R-10087</td>
<td>The VNF package <strong>MUST</strong> contain all standard artifacts as specified in ETSI GS NFV-SOL004 including Manifest file, VNFD (or Main TOSCA/YAML based Service Template) and other optional artifacts. CSAR Manifest file as per SOL004 - for example ROOT*MainServiceTemplate.mf</td>
<td>Shall applicable to PNF package</td>
</tr>
<tr>
<td>R-01123</td>
<td>The VNF package Manifest file <strong>MUST</strong> contain: VNF package meta-data, a list of all artifacts (both internal and external) entry’s including their respected URI’s, an algorithm to calculate a digest and a digest result calculated on the content of each artifacts, as specified in ETSI GS NFV-SOL004. The VNF Package MUST include VNF Identification Data to uniquely identify the resource for a given VNF provider. The identification data must include: an identifier for the VNF, the name of the VNF as was given by the VNF provider, VNF description, VNF provider, and version.</td>
<td>Shall applicable to PNF package With new valid names/values - pnf_provider_id - pnf_product_name - pnf_release_date_time - pnf_package_version</td>
</tr>
<tr>
<td>R-21322</td>
<td>The VNF provider <strong>MUST</strong> provide their testing scripts to support testing as specified in ETSI NFV-SOL004 - Testing directory in CSAR</td>
<td>Should applicable to PNF package</td>
</tr>
</tbody>
</table>
| R-26885 | The VNF provider **MUST** provide the binaries and images needed to instantiate the VNF (VNF and VNFC images) either as:  
  - Local artifact in CSAR: ROOT\Artifacts\VNF\image.bin  
  - externally referred (by URI) artifact in Manifest file (also may be referred by VNF Descriptor)  
**Note:** Currently, ONAP doesn’t have the capability of Image management, we upload the image into VIM/VNFM manually. | May applicable to PNF package Not supported with current release |
| R-40820 | The VNF provider **MUST** enumerate all of the open source licenses their VNF(s) incorporate. CSAR License directory as per ETSI SOL004. for example ROOT\Licenses\License_term.txt | May applicable to PNF package |
| R-xxxx | **VNF Package Authenticity** | May applicable to PNF package |
## PNF on-boarding requirements (7.2)

<table>
<thead>
<tr>
<th>R number</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-77707</td>
<td>The xNF provider <strong>MUST</strong> include a Manifest File that contains a list of <strong>all</strong> the components in the xNF package</td>
<td>OK. Overlapped with R-10087 in section 5.1.6.3</td>
</tr>
<tr>
<td>R-66070</td>
<td>The xNF Package <strong>MUST</strong> include xNF Identification Data to uniquely identify the resource for a given xNF provider. The identification data must include: an identifier for the xNF, the name of the xNF as was given by the xNF provider, xNF description, xNF provider, and version.</td>
<td>Part of the descriptor</td>
</tr>
<tr>
<td>R-98617</td>
<td>The xNF provider <strong>MUST</strong> provide information regarding any dependency (e.g., affinity, anti-affinity) with other xNFs and resources.</td>
<td>Part of the descriptor</td>
</tr>
<tr>
<td>R-22346</td>
<td>The VNF package <strong>MUST</strong> provide VES Event Registration for all VES events provided by that xNF.</td>
<td>VES event Registration Should be applicable to PNF</td>
</tr>
<tr>
<td>R-89571</td>
<td>The xNF <strong>MUST</strong> support and provide artifacts for configuration management using at least one of the following technologies; a) Netconf/YANG, b) Chef, or c) Ansible.</td>
<td></td>
</tr>
<tr>
<td>R-30278</td>
<td>The xNF provider <strong>MUST</strong> provide a Resource/Device YANG model as a foundation for creating the YANG model for configuration. This will include xNF attributes/parameters and valid values/attributes configurable by policy.</td>
<td>Not the proposed FM dictionary</td>
</tr>
<tr>
<td>R-27711</td>
<td>The xNF provider <strong>MUST</strong> provide an XML file that contains a list of xNF error codes, descriptions of the error, and possible causes/corrective action</td>
<td>Not supported by Casablanca</td>
</tr>
<tr>
<td>R-74763</td>
<td>The xNF provider <strong>MUST</strong> provide an artifact per xNF that contains all of the xNF Event Records supported. The artifact should include reference to the specific release of the xNF Event Stream Common Event Data Model document it is based on. (e.g., VES Event Listener)</td>
<td>VES event Listener</td>
</tr>
<tr>
<td>R-35851</td>
<td>The xNF Package <strong>MUST</strong> include xNF topology that describes basic network and application connectivity internal and external to the xNF including Link type, KPIs, Bandwidth, latency, jitter, QoS (if applicable) for each interface.</td>
<td>Part of the descriptor?</td>
</tr>
<tr>
<td>R-26881</td>
<td>The xNF provider <strong>MUST</strong> provide the binaries and images needed to instantiate the xNF (xNF and VNFC images).</td>
<td></td>
</tr>
<tr>
<td>R-96634</td>
<td>The xNF provider <strong>MUST</strong> describe scaling capabilities to manage scaling characteristics of the xNF.</td>
<td>Not supported by PNF</td>
</tr>
<tr>
<td>R-04298</td>
<td>The xNF provider <strong>MUST</strong> provide their testing scripts to support testing.</td>
<td>Testing Requirements.</td>
</tr>
<tr>
<td>R-58775</td>
<td>The xNF provider <strong>MUST</strong> provide software components that can be packaged with/near the xNF, if needed, to simulate any functions or systems that connect to the xNF system under test. This component is necessary only if the existing testing environment does not have the necessary simulators.</td>
<td></td>
</tr>
<tr>
<td>R-85653</td>
<td>The xNF <strong>MUST</strong> provide metrics (e.g., number of sessions, number of subscribers, number of seats, etc.) to ONAP for tracking every license.</td>
<td>Only if Licensing is needed</td>
</tr>
<tr>
<td>R-40827</td>
<td>The xNF provider <strong>MUST</strong> enumerate all of the open source licenses their xNF(s) incorporate.</td>
<td></td>
</tr>
<tr>
<td>R-85991</td>
<td>The xNF provider <strong>MUST</strong> provide a universal license key per xNF to be used as needed by services (i.e., not tied to a VM instance) as the recommended solution. The xNF provider may provide pools of Unique xNF License Keys, where there is a unique key for each xNF instance as an alternate solution. Licensing issues should be resolved without interrupting in-service xNFs.</td>
<td></td>
</tr>
<tr>
<td>R-47849</td>
<td>The xNF provider <strong>MUST</strong> support the metadata about licenses (and their applicable entitlements) as defined in this document for xNF software, and any license keys required to authorize use of the xNF software. This metadata will be used to facilitate onboarding the xNF into the ONAP environment and automating processes for putting the licenses into use and managing the full lifecycle of the licenses.</td>
<td></td>
</tr>
</tbody>
</table>
Q&A
VNF SDK Updates

Benjamin Cheung, PhD
## TOSCA Artifacts

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tosa Model</td>
<td>TOSCA_CSAR</td>
<td>0</td>
</tr>
<tr>
<td>Tosa Template</td>
<td>TOSCA_TEMPLATE</td>
<td>0</td>
</tr>
</tbody>
</table>
VNF SDK Impacts

• **VNFKSDK-337**: Supporting PNF package onboarding
  - **VNFKSDK-338**: Project scope to include PNF
  - **VNFKSDK-339**: PNF CSAR structure based SOL004
  - **VNFKSDK-340**: PNF manifest file
  - **VNFKSDK-341**: PNFD validation based on SOL001
  - **VNFKSDK-342**: Support packaging security
  - **VNFKSDK-343**: Enhancement of the test on PNF package
Design time view of a service
Internal representation of a model
Onboarding Model > SDC produces
Platform (Internal) Information Model

NF

ANF  PNF  VNF

Capabilities  Requirement  Capabilities  Requirement

Create “PNF”
Assign (registers)/connpts
Configure

Assign [connections pts]
Create
Configure
5G RAN Wireless Systems

Benjamin Cheung, PhD
5G Key Technology Components

- New Spectrum (Rel 15, 52.6 GHz/39 GHz, Rel 16 > 52.6 GHz)
- Advanced Beamforming
- Multi-Connectivity (NSA, SA, Option 3, 4, 7)
- Network Slicing
- Edge Computing

- Software Defined Networking (SDN)
- Network Functions Virtualization (NFV)
- Fog Computing (FC)
- Mobile Edge Computing (MEC)
5G RAN Wireless Systems & ONAP

Benjamin Cheung, PhD
5G RAN Wireless Network

**RAN Network Elements**
- **Antenna**
- **RU** – Remote Radio Unit
- **DU** – Distributed Unit (5G Base Unit)
- **CU** – Centralized Unit

**Core Network Elements**
- **Disaggregated Core**
  - UPF – User Plane Function
  - SMF – Session Management Function
  - UDM – Unified Data Management Function
  - AUSF – Authentication Service Function

**ONAP Run Time Management**
- **OSS BSS**
- **SO**
- **SDN-C**
- **DCAE**
- **A&AI**
- **APP-C**

SO – Service Orchestrator
SDN-C – Service Design Network Controller
DCA&E – Data Collection Analytics & Events
A&AI – Available & Active Inventory
APP-C – Application Control

**5G Application Ecosystem**
- **UE**
- **ONAP Run Time Management**

**RAN Network Elements**
- **Edge Cloud**
- **Centralized Cloud**
- **External Content**

**Core Network Elements**
- **Internet**

**Notes:**
- RU – Remote Radio Unit
- DU – Distributed Unit (5G Base Unit)
- CU – Centralized Unit
R4: Modeling a 5G Service

5G SERVICE

- Informational Artifacts
  - Configuration Files
- Deployment Artifacts
  - HEAT Templates
  - Connection Point
  - Virtual Link
- User Designed
  - Policies
  - Templates
  - Blueprints
- Foundational
- Network Function (VNF)
  - Network Function PNF
  - PNF Device
- Resources

Vendor Provided

- Foundational
- User Designed

THE LINUX FOUNDATION
R4: 5G Base Station (gNodeB)
Configurations

Benjamin Cheung, PhD
MODELING WITHIN A PNF (DU)

5G DU (PNF)

- NF PNF – “Application” S/W does 5G voice/data
- N/F Node Type
- Hardware PNF - Modem (5G DU) [Hardware]
- H/W Node Type

Sub-component #1

Sub-component #2

Sub-component #n

SFP #1 = Port #1

SFP #n = Port #n

Software Function of a DU

Network Function

“Hardware Aspects of a PNF”

Connection Point

Ports / NICs

The hardware Ports (e.g. SFP/Backhaul Ports)

Sub-components within PNF

NF Component

FUTURE

H/W Node Type

NF PNF – Hardware PNF

The Linux Foundation
DU Configurations

DRIVE PARTITIONS

5G DU (PNF)
Partition1 (Active) Version 2018-09A
Partition2 (Inactive) Version 2018-09B

SUBCOMPONENTS (R4+)

PNF Version 2018-09A
Sub-Component-1 Version 2018-09A
Sub-Component-2 Version 2018-09B

MULTI-PNF DAISY CHAIN CONFIG

Version 2018-09A

TANDEM CHASSIS CONFIGURATIONS

PNF
Version LTE2018-09A
Sub-Frame DU#1 Version 2018-09B
Sub-Frame DU#2 Version 2018-09B
Sub-Frame DU#2
ETSI (SOL 001, SOL 004, SOL005, SOL 007)
## VNFD/PNFD/NSD Properties Alignment

**Comments**

<table>
<thead>
<tr>
<th>PNFD</th>
<th>VNFD</th>
<th>NSD</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>descriptor_id</td>
<td>descriptor_id</td>
<td>descriptor_id</td>
<td>Proposed new name in PNFD: descriptor_version</td>
</tr>
<tr>
<td>version</td>
<td>descriptor_version</td>
<td>descriptor_version</td>
<td>Better to be aligned. Proposed new name in VNFD: name</td>
</tr>
<tr>
<td>provider</td>
<td>provider</td>
<td>provider</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>product_info_name</td>
<td>name</td>
<td></td>
</tr>
<tr>
<td>software_version</td>
<td>software_version</td>
<td>software_version</td>
<td>New in PNFD 1. Better to align with VNFD. VNFD uses software_version for only software change. descriptor_id might be changed only due to descriptor evolution itself like security adding. 2. Long-term view, it would be useful to upgrade PNF software. 3. It is also useful for service provider to get such information for OAM view like trouble-shooting, service checking, PNF packability checking and so on 4. Align with ONAP model</td>
</tr>
<tr>
<td>function_description</td>
<td>product_info_description</td>
<td>-</td>
<td>Proposed new name in VNFD: function_description. Add function_description into NSD</td>
</tr>
<tr>
<td>descriptor_invariant_id</td>
<td>product_name</td>
<td>invariant_id</td>
<td>Proposed new name in VNFD / PNFD: invariant_id</td>
</tr>
<tr>
<td>geographical_location_info</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

On-going ETSI discussion
Tentative updated SOL004 title:
“Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; VNF Package specification and PNF file specification”

Tentative updated scope:
The present document specifies the structure and format of a VNF package file and its constituents, fulfilling the requirements specified in ETSI GS NFV-IFA 011 [1] for a VNF package and in ETSI GS NFV-IFA 014 [x] for a PNFD.

Working schedule:

<table>
<thead>
<tr>
<th>Milestone name</th>
<th>Target date</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR approval with expanded scope of 2.6.1</td>
<td>2018/12/07</td>
</tr>
<tr>
<td>Functional CRs approved</td>
<td>2018/12/13</td>
</tr>
<tr>
<td>WG approval</td>
<td>2018/12/31</td>
</tr>
<tr>
<td>TB approval</td>
<td>2019/01/31</td>
</tr>
</tbody>
</table>

On-going ETSI discussion
## Deployment Artifact

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Deployment timeout</th>
<th>Version</th>
<th>UUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>VF License</td>
<td>VF_LICENSE</td>
<td></td>
<td>1</td>
<td>6e2400-c-a2de-483f-acc1-c31acba7e4f6</td>
</tr>
<tr>
<td>basic_vftv</td>
<td>HEAT</td>
<td>60</td>
<td>2</td>
<td>1622a222-0002-4468-979e-239f46ed8e6f4f</td>
</tr>
<tr>
<td>VF HEAT ENV</td>
<td>HEAT_ENV</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Vendor License</td>
<td>VENDOR_LICENSE</td>
<td></td>
<td>1</td>
<td>b8c223ba-50b7-46ee-9182-af9389491239</td>
</tr>
</tbody>
</table>

---
networks:
- network: { get_param: public_net_id }
- port: { get_resource: vfw_private_0_port }
- port: { get_resource: vfw_private_1_port }
- port: { get_resource: vfw_private_2_port }
metadata: (vnf_id: { get_param: vnf_id }, vf_module_id: { get_param: vf_module_id })
user_data_format: RAW
user_data:
str_replace:
  param:
  __dcae_collector_ip__: { get_param: dcae_collector_ip }
  __dcae_collector_port__: { get_param: dcae_collector_port }
  __repo_url_blob__: { get_param: repo_url_blob }
  __repo_url_artifacts__: { get_param: repo_url_artifacts }
  __demo_artifacts_version__: { get_param: demo_artifacts_version }

def:
  DCAE_COLLECTOR_IP= __dcae_collector_ip__
  DCAE_COLLECTOR_PORT= __dcae_collector_port__
  REPO_URL_BLOB= __repo_url_blob__
  REPO_URL_ARTIFACTS= __repo_url_artifacts__
  DEMO_ARTIFACTS_VERSION= __demo_artifacts_version__

# Download required dependencies
add-apt-repository -y ppa:openjdk-r/ppa
apt-get update
apt-get install -y make wget openjdk-8-jdk gcc libcurl14
cd /opt

mvn org.apache.maven.plugins:maven-dependency-plugin:2.10: get -DremoteRepositories=$REPO_URL_ARTIFACTS
-Dtransitive=false -Ddest=
-Dtransitive=false -Ddest=
tar -zxvf ves-$DEMO_ARTIFACTS_VERSION-demo.tar.gz
mv ves-$DEMO_ARTIFACTS_VERSION VES
mv ves-$DEMO_ARTIFACTS_VERSION-VESreporting-VFM
tar -zxvf sample-distribution-$DEMO_ARTIFACTS_VERSION-VESreporting-VFM:
mv sample-distribution-$DEMO_ARTIFACTS_VERSION honeycomb